

Mobile Computing's Impact on Software Construction

Wolfgang Pree
University of Constance/UC Berkeley
pree@eecs.berkeley.edu
www.SoftwareResearch.net

Contents

- Motivation
- Adaptive software components
- Selected research projects



Motivation

Where is computing headed?

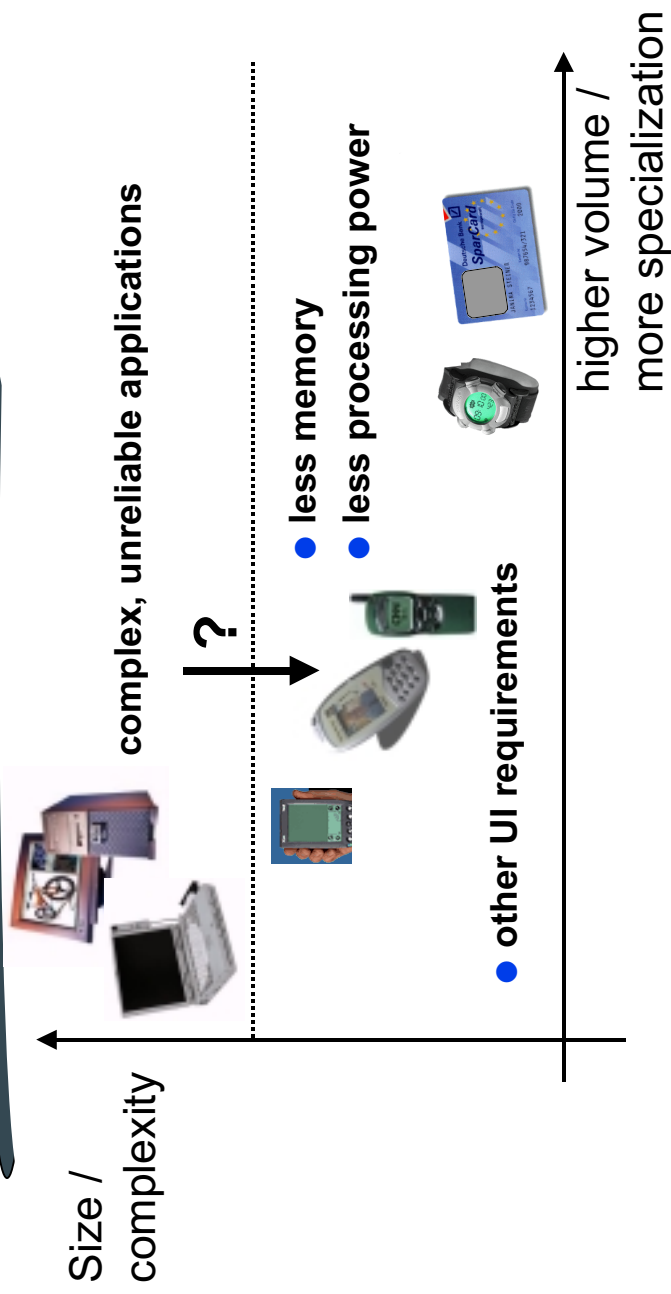


PCs as **one-size-fits-all** devices are **too complicated** and **too clumsy** for many users.

=> trend towards specialized gadgets

- PDAs: PalmPilot, Psion, Casio, etc.
- digital books and encyclopedias
- set-top box + TV= Web access
- chip cards
- pen-like text scanners
- . . .

Applications on mobile gadgets require a redesign



Relevant research directions (I)

Negotiating, adaptive software components

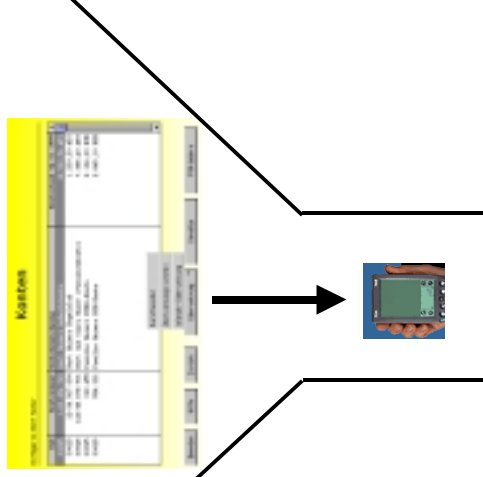
- domain descriptions
- consideration of geographic position
- development and testing
- . . .

potential scenarios assume:

- **ad-hoc component coupling** for accomplishing
- **simple tasks**, eg printing/displaying documents
(→ Jini)
analogous to fax machines which negotiate transmission speed and the common protocol for the task *fax transmission*

Relevant research directions (II)

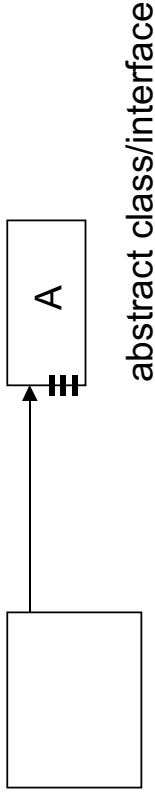
- component technology/
software architecture,
in particular for
distributed systems &
embedded systems
- human-computer
interaction aspects



Adaptive software components

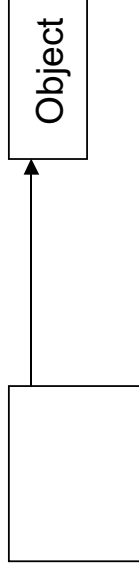
Reflection forms the basis of negotiating components

Abstract coupling represents the traditional construction principle in OO to provide plug & work of components:



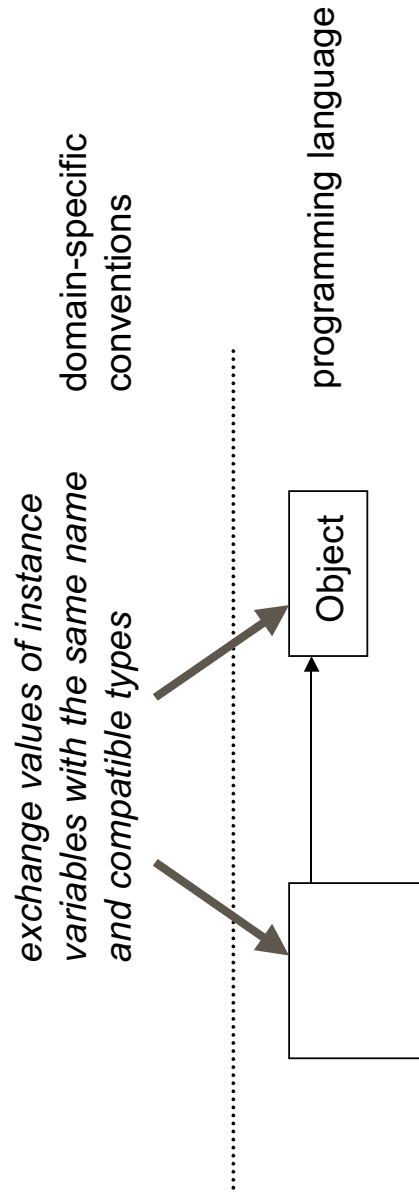
If any object offers meta-information, a component can be coupled with the most general type, typically called

Object:



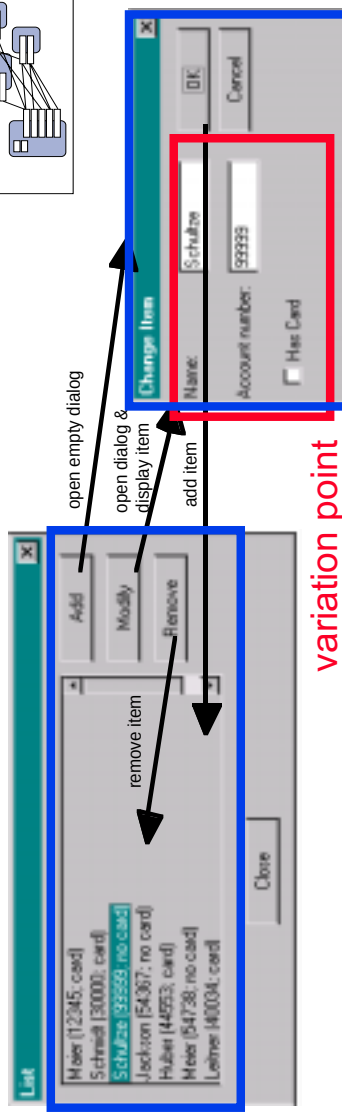
Coupling semantics have to be defined separately

The simplest way to define coupling semantics is a naming convention

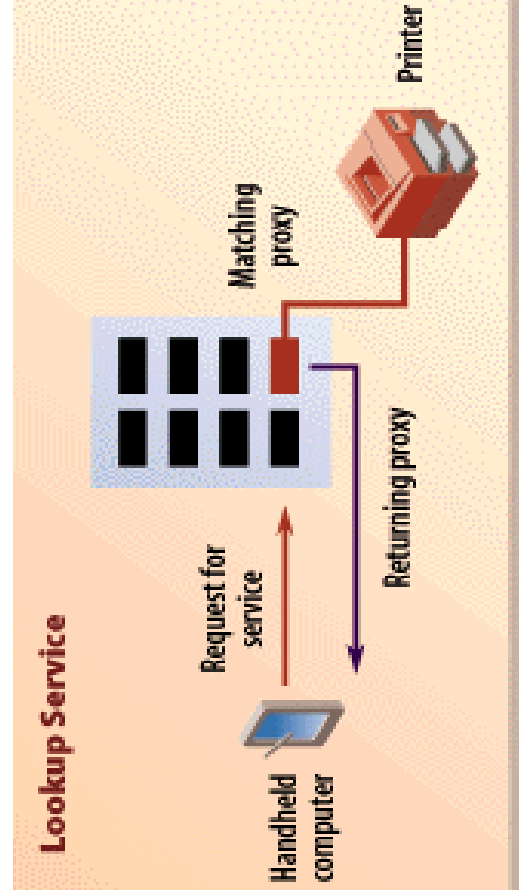


Example: ListX framelet (JavaBean)

Automating the data exchange between any (inner) dialog and any list item:



Finding services in Jini is also based on naming conventions



More advanced concepts better support ad-hoc coupling



- Dynamic adapters
- Coordination frameworks

Dynamic adapters (I)



A standardized domain description, eg, comprising associated services, forms the basis for coupling components which do not know each other.

Dynamic adapters allow a collaboration of components which were developed independently of others.

GoF Adapter	Dynamic Adapter
static translation	dynamic translation



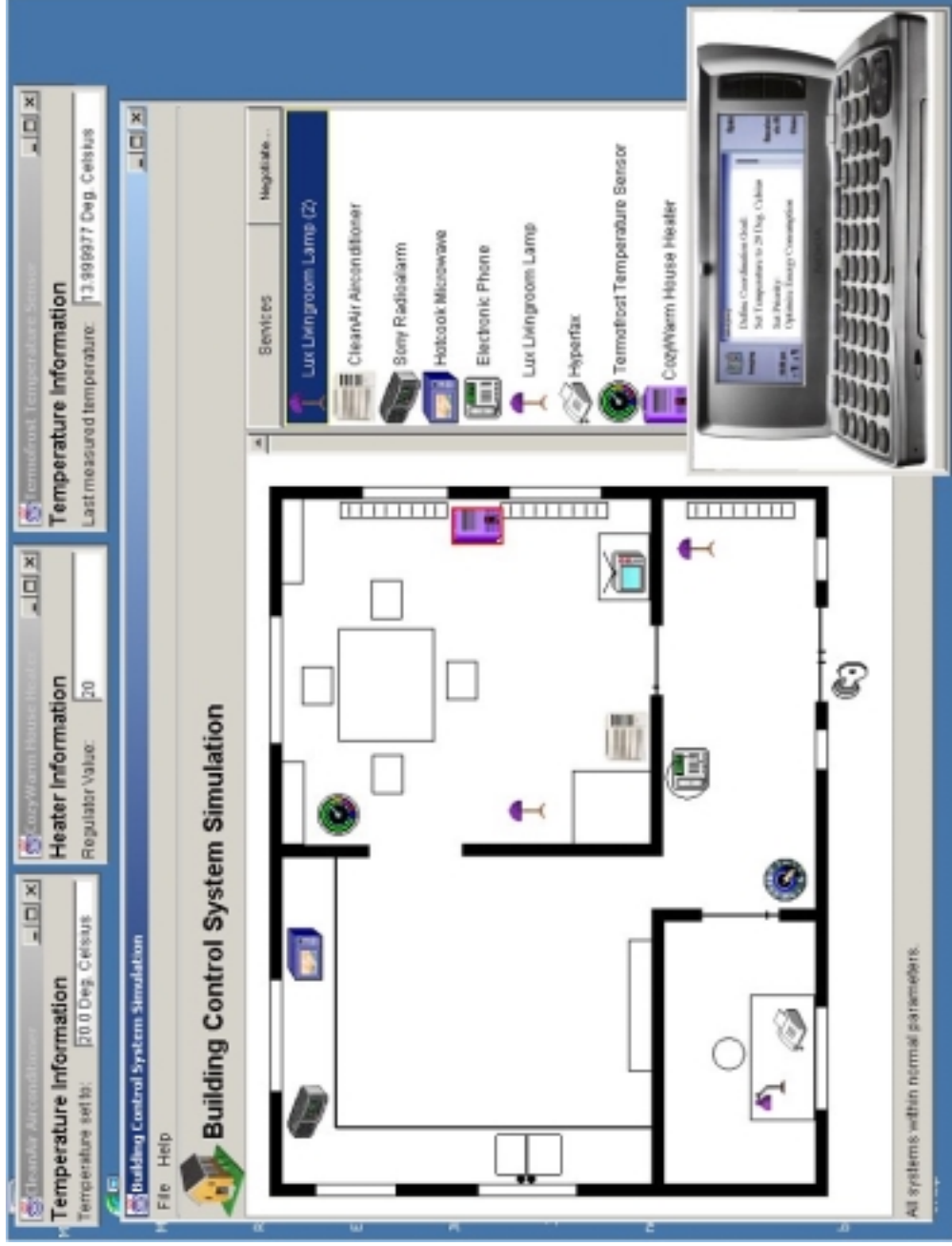
Selected research projects

MoCCA (Mobile Communication & Computing Applications)

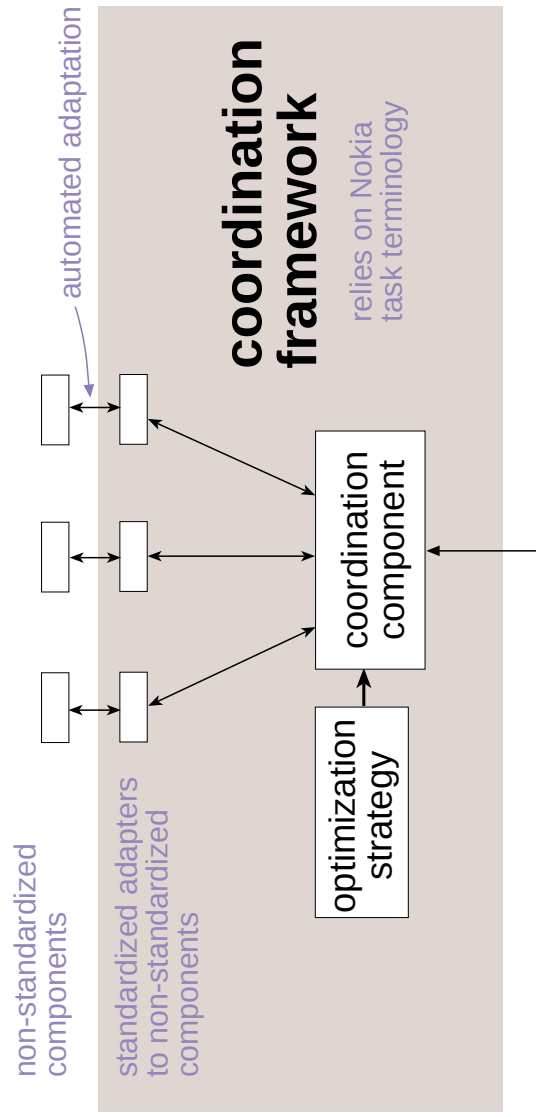


Goals:

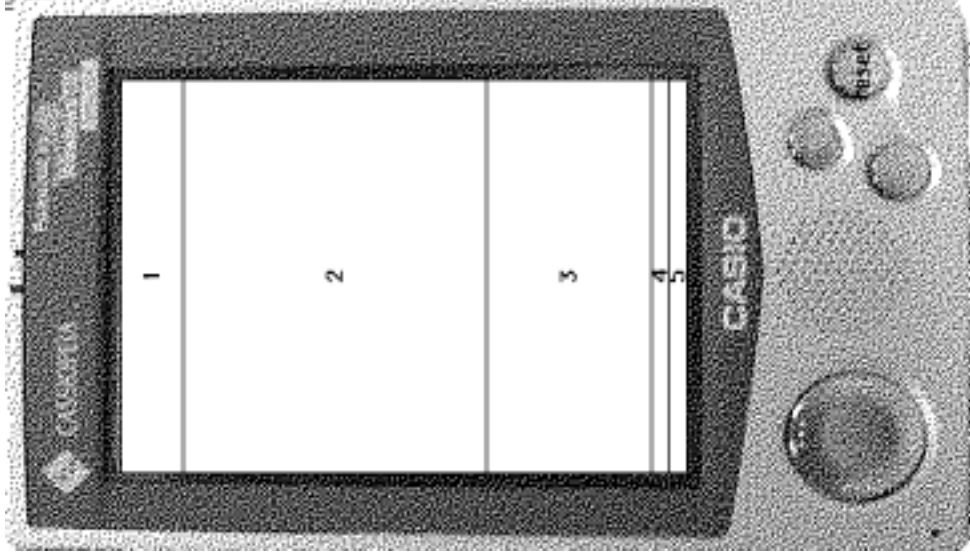
- explore application domains of mobile/cellular computing;
- serve as test bed for research in the areas adaptive components and human-computer interaction

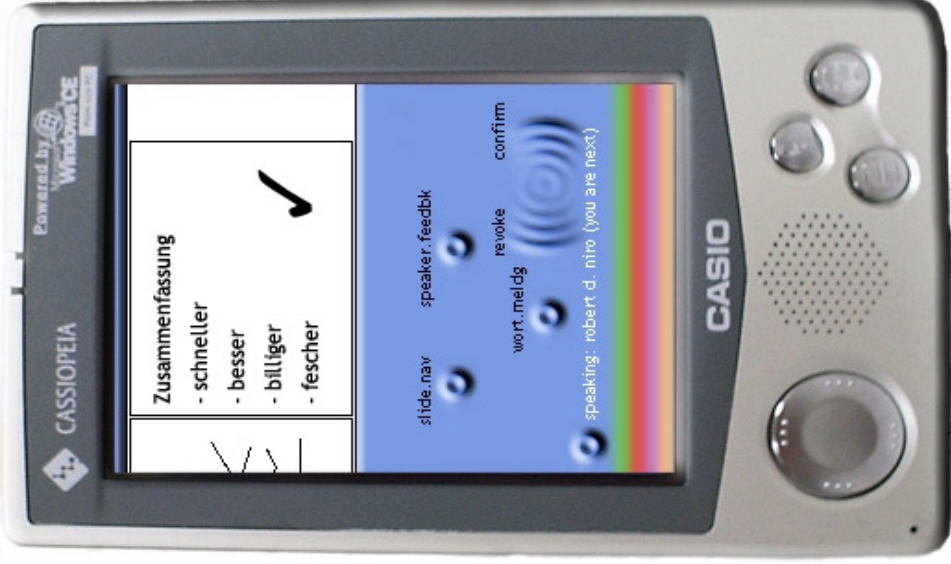


Dynamic adapters (II)

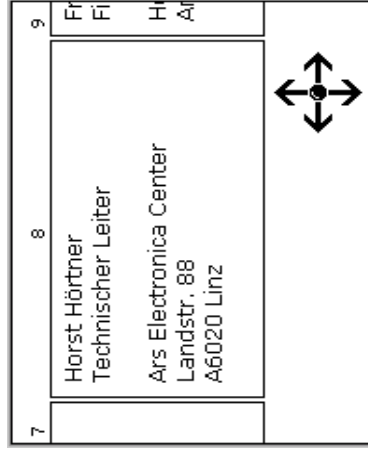


Tiled application
windows
resizable with your
thumb

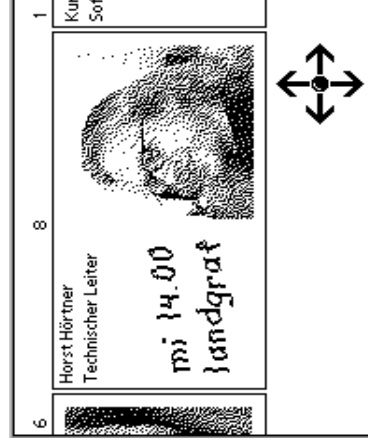


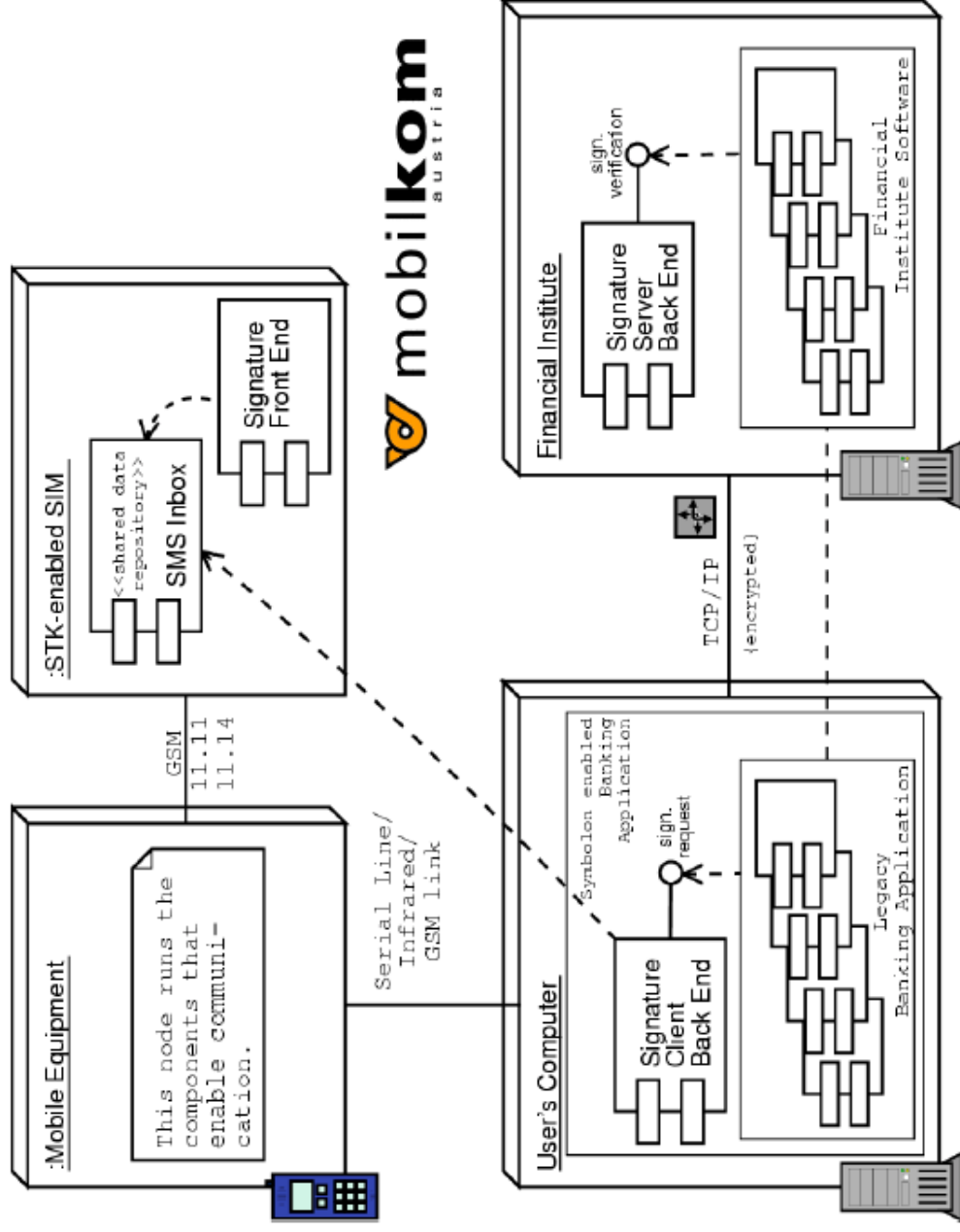


Navigating a two-dimensional infospace

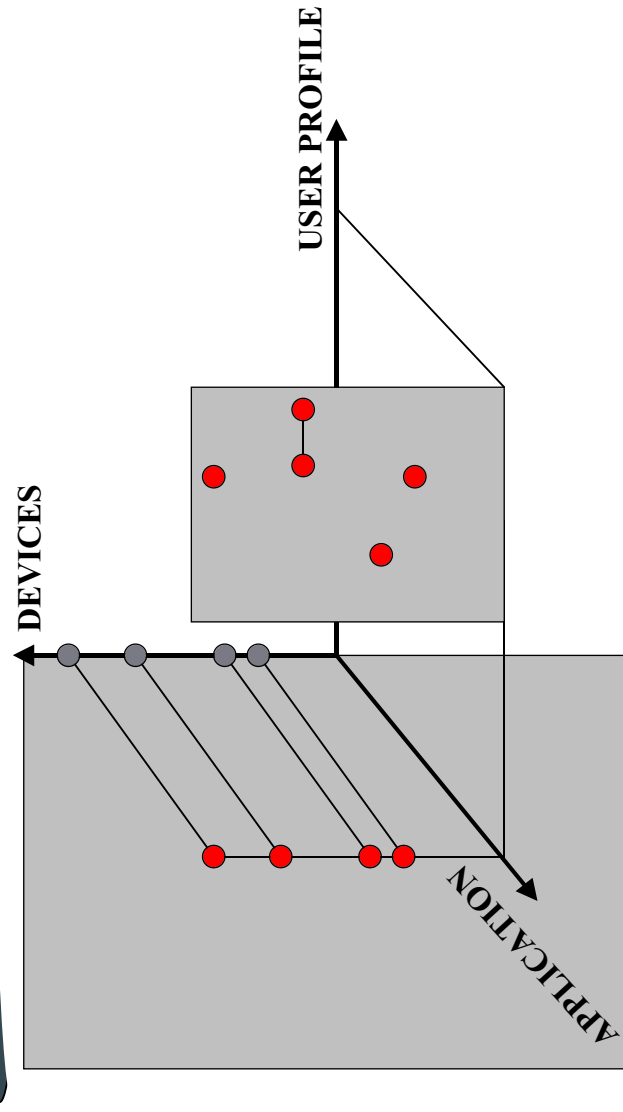


up
↑

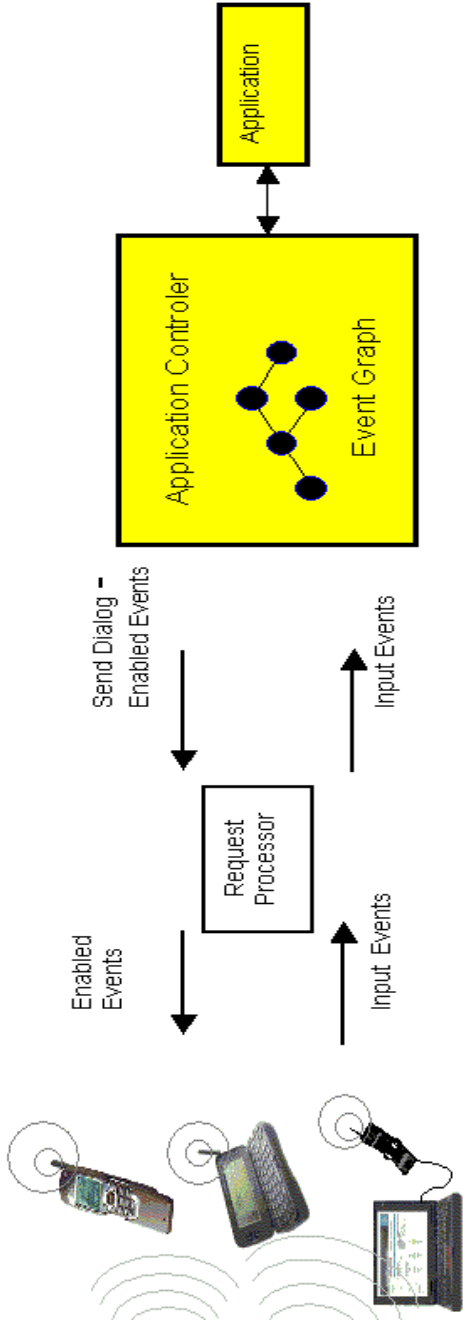




Adaptive Frontend Architecture (I)

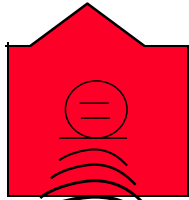


Adaptive Frontend Architecture (II)



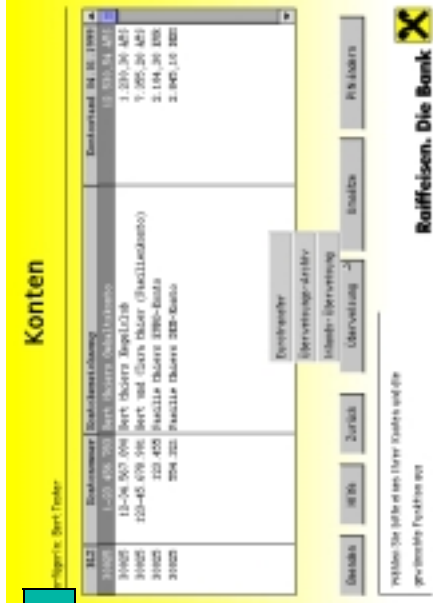
Voice-controlled Internet banking

Java-DDE
(ca. 10 KB)



Überweisung
Umsätze
...

Philips VoCon
(ca. 100 KB)



Voice-controlled Euro conversion in a watch



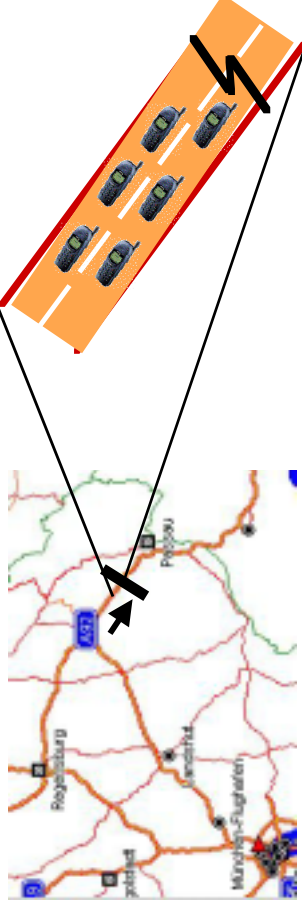
Four
point
two
three

Euro

Applications based on cell phones

Cell phones with GPS-like positioning capabilities.

- immediate identification of traffic jams => warning of those who are on the way there



- measuring the popularity of skiing resorts, eg, on weekends

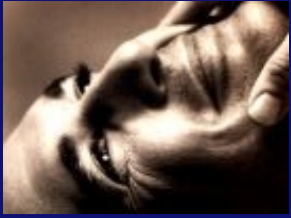


Conclusions



Conclusions

- New mobile devices have a significant impact on application development
- Network-based software architectures are emerging



The race belongs to the swift